No Lecture Materials
or Mobile Devices

90min, 100 points

TECHNISCHE UNIVERSITÄT BERLIN

Architecture of ML Systems (AMLS)
12b Q&A and Exam Preparation [continues 5.45pm]

Prof. Dr. Matthias Boehm
Technische Universität Berlin
Berlin Institute for the Foundations of Learning and Data
Big Data Engineering (DAMS Lab)

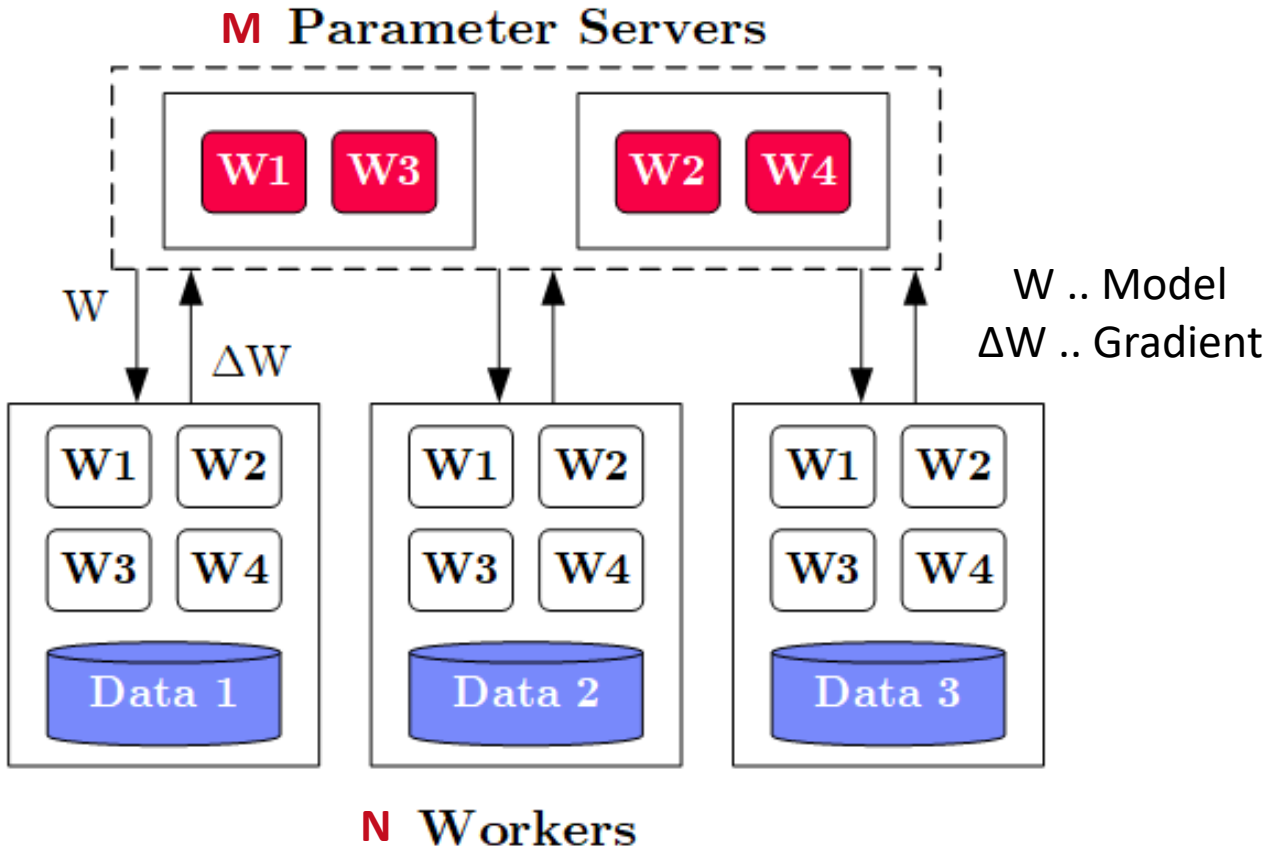
Last update: Jul 12, 2024

BIFOLD

Task 1 Parameter Servers [question appeared in every exam]

- **Task 1a: Describe the overall system architecture of data-parallel parameter servers, explain its components and interaction among these components [10/100 points]**

- **System Architecture**
  - **M** Parameter Servers w/ model
  - **N** Workers w/ data partitions
  - Optional Coordinator
- **Interactions**
  - Workers **pull** model from parameter servers, slice a mini-batch of data, run a forward and backward pass to compute gradients, which are **pushed** back to parameter serves
  - Parameter servers wait for gradients, **aggregate** the gradients/models, and perform a **global model update**



Task 1 Parameter Servers, cont. [question appeared in every exam]

- **Task 1b: Describe synchronous (BSP) and asynchronous (ASP) update strategies in data-parallel parameter servers and name their advantages and disadvantages. [6/100 points]**

	Synchronous	Asynchronous
Description	Per-batch or -n-batches synchronization barrier (wait for all workers before update)	Every pushed gradient updates the model, workers obtain model immediately
Advantages	**Consistent learning** process and model updates	**No waiting** for stragglers
Disadvantages	**Workers wait** for slowest worker (repeatedly)	Workers use stale models, **potential divergence**

Task 2 Data Preparation

- **Task 2a: Given the raw input data below, apply recoding and one-hot encoding to all categorical columns, and binning with 3 equi-width bins to all numerical columns. [10/100 points]**

A	B	C
Low	0	S
High	3	M
Med	7	L
Low	9	XL
Low	15	M
Low	7	M
Med	4	L
High	12	XL
High	13	L

ALow	AMed	AHigh	B	CS	CM	CL	CXL
1	0	0	1	1	0	0	0
0	0	1	1	0	1	0	0
0	1	0	2	0	0	1	0
1	0	0	2	0	0	0	1
1	0	0	3	0	1	0	0
1	0	0	2	0	1	0	0
0	1	0	1	0	0	1	0
0	0	1	3	0	0	0	1
0	0	1	3	0	0	1	0

Task 2 Data Preparation, cont.

- **Task 2b: What is feature hashing and what is its advantage over recoding? [3/100 points]**
  - **Hash values** and compute modulo with **user-provided k**
  - Reduces the number of distinct items, and thus columns in one-hot-encoded representation

- **Task 2c: Describe the text encodings bag-of-word and word-embeddings. [6/100 points]**
  - **Bag-of-word:** encode sentence as a **vector of token counts** (how often every distinct token appeared in the sentence)
  - **Word Embedding:** continuous bag-of-words, learned numerical vectors for **predicting the context words** from a word or a word from its context

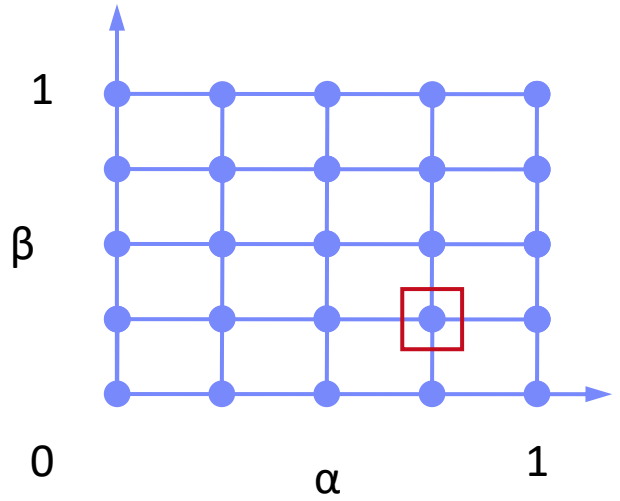
A	B	C	D	E
2	2	1	0	1

- **Task 2d: What is data augmentation and name 2 concrete techniques. [3/100 points]**
  - **Synthetically generate** labeled examples from small real labeled dataset through transformations
  - **Examples:** rotations, reflections, shearing, noise modulation

Task 3 Model Selection

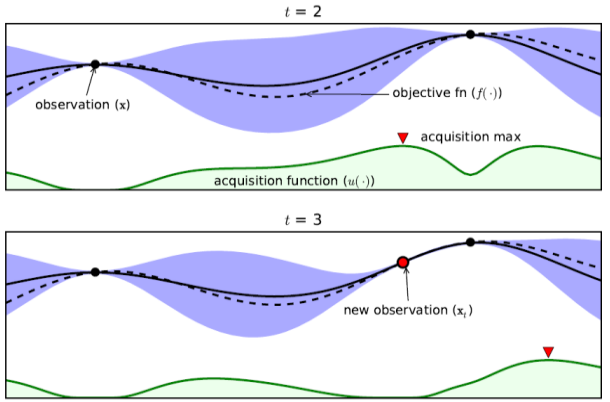
- **Task 3a: Describe the task of hyper-parameter tuning by example of GridSearch. Assume three hyper-parameters with 10 discretized values each, how many models do we need to train? [8/100 points]**

- **Hyper Parameter Tuning**
  - Given a model and dataset, **find best hyper parameter values** (e.g., learning rate, regularization, kernel parameters, tree params) by training the model and evaluating it on the validation set.
- **Grid Search**
  - Discretize continuous parameters (linearly or exponentially)
  - Every hyper-parameter is a dimension of a hyper-cube
  - For all combinations, train and evaluate the model
- **Example: 10^3 = 1000 trained models**



Task 3 Model Selection, cont.

- **Task 3b: Explain Bayesian Optimization as a more directed search strategy, and how it balances exploitation and exploration? [5/100 points]**
  - Use **lightweight ML** models like Gaussian Processes to find next points
  - **Acquisition function** to balance exploitation (expected mean) and exploration (uncertainty, expected variance)



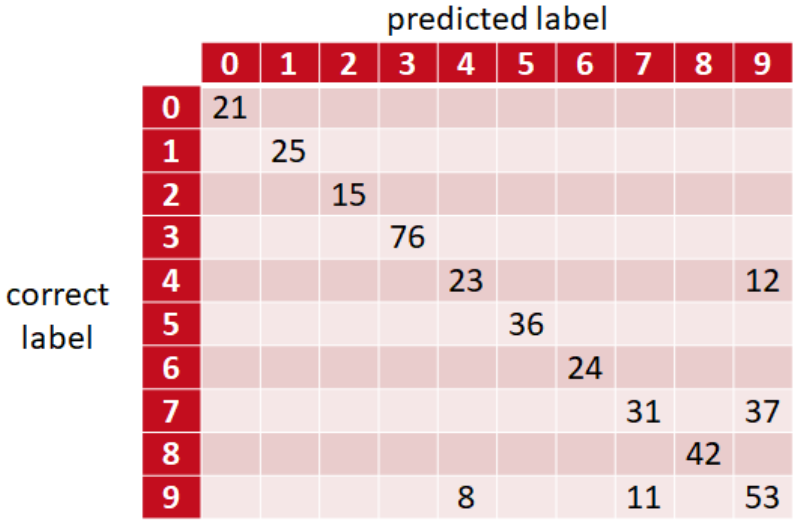
- **Task 3c: Describe the problem of neural architecture search, and how to deal with multiple optimization objectives (e.g., accuracy and runtime). [5/100 points]**
  - Automatically compose neural network architectures from building blocks
  - **Search strategies: evolutionary algorithms** and bayesian optimization
  - **Multi-objective optimization:** (1) **linearization**, (2) **pareto front to user**, (3) **primary objective w/ constraints** on other dims

Task 4 Model Debugging

64/100

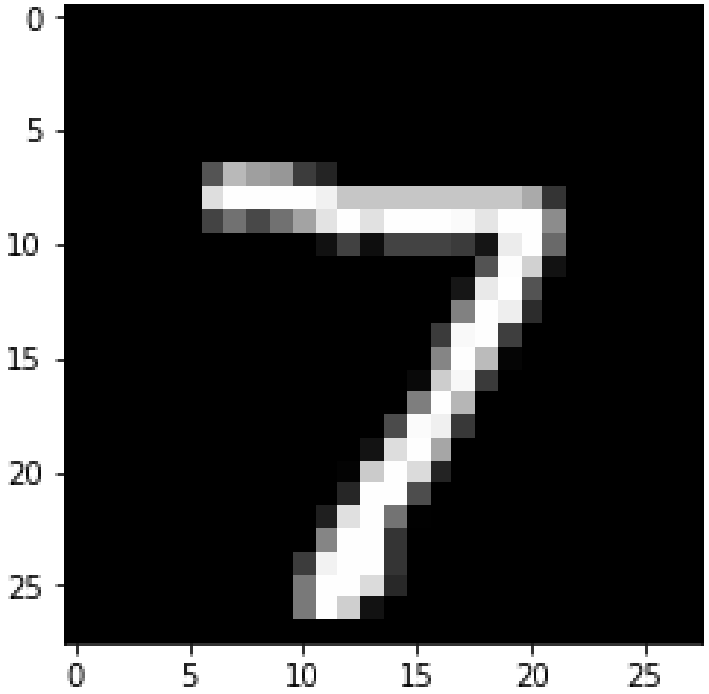
- **Task 4a: Describe sources of bias in machine learning and name examples how to ensure fairness when building ML models with examples. [4/100 points]**
  - **Sources:** selection bias, sample bias, data bias (e.g., NMAR), confirmation bias
  - **Fairness constraints:** monotonicity, group fairness constraints

- **Task 4b: Explain the concept of a confusion matrix and describe it in detail. [4/100 points]**
  - **Matrix of correct versus predicted labels**
  - Cells contain counts or relative frequencies of correct/predicted pair occurrences
  - Enables understanding which classes are "confused" with each other

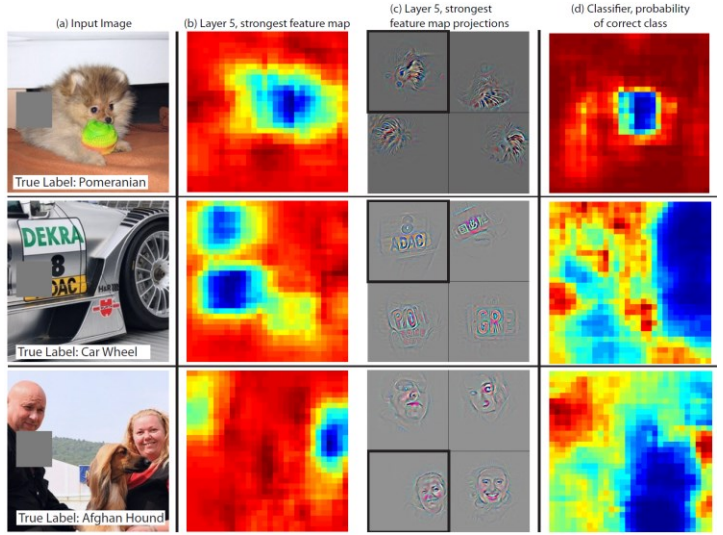


Task 4 Model Debugging, cont.

- **Task 4c: Explain the concept of occlusion-based explanations by example of classifying below hand-written digit as a seven [4/100].**

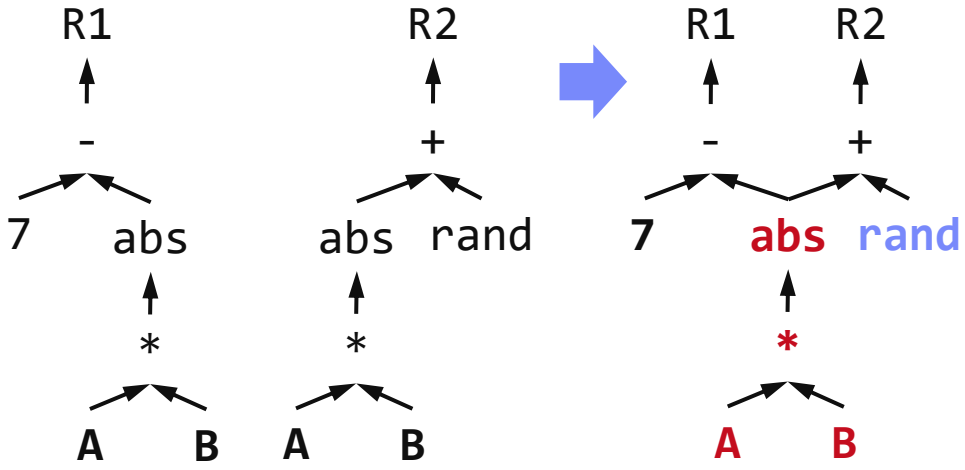


- **Slide black/gray square over input image**
- **Measure how feature maps (layer activation) and classifier output change**
- **Show a heat map of these changes**



Task 5 Compilation Techniques

- **Task 5a: Describe the purpose of the rewrite common subexpression elimination (CSE) and sketch an algorithm to perform CSE on a directed acyclic graph (DAG) of operators. [5/100 points]**
  - Convert tree/graph of operators with redundant common subexpressions into redundancy-free operator graph
  - **Step 1:** Collect and **replace leaf nodes** (variable reads and literals)
  - **Step 2:** recursively **remove CSEs bottom-up** starting at the leaves by merging nodes with same inputs (**beware non-determinism**)

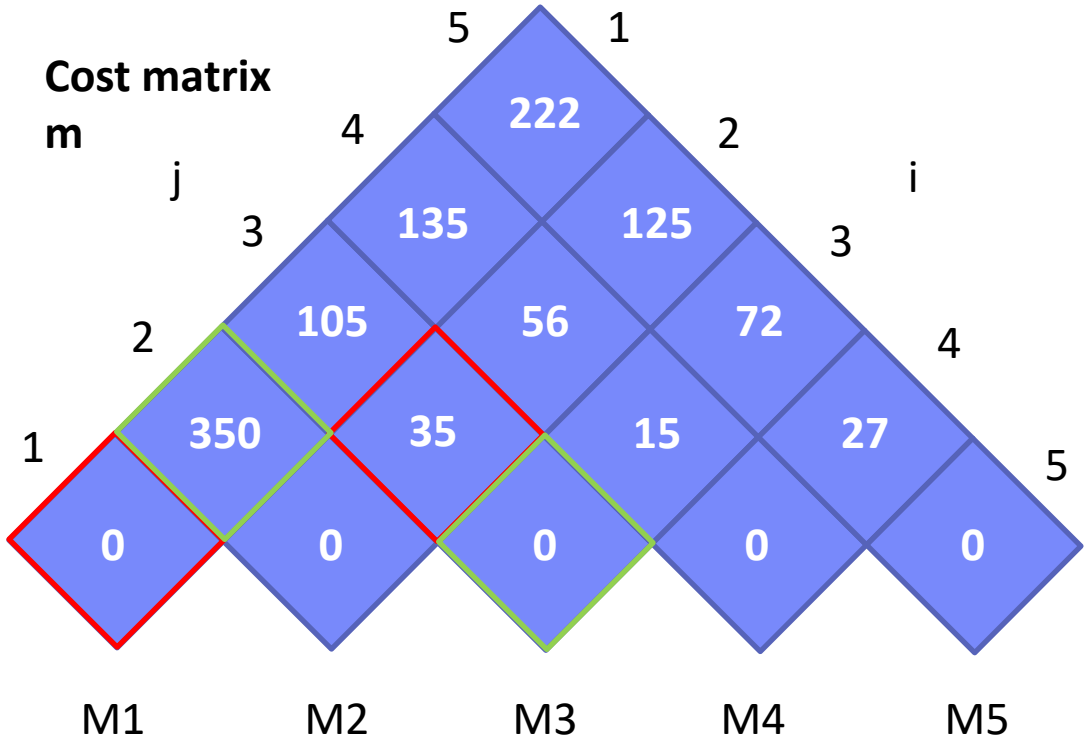


- **Task 5b: Explain the concept of operator fusion and how it can improve runtime performance [3/100 points]**
  - Merge sequence or sub-DAG of data-dependent operators into a single operator
  - **Performance Improvements:** **avoid unnecessary allocation**, reduced write/read **memory bandwidth** requirements / cache locality, **additional specialization** (e.g., data types, dimensions)

Task 5 Compilation Techniques

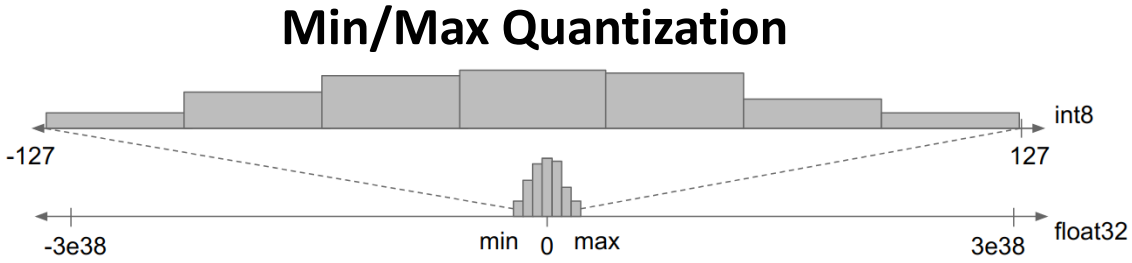
- **Task 5c: Assume an example chain of matrix multiplications (A B C D E), describe the problem of matrix multiplication chain optimization, and a dynamic programming algorithm for solving it efficiently [7/100 points]**

- **Matrix Multiplication is associative, mmchain opt aims to find optimal parenthesization**
- **Dynamic programming applies because (1) optimal substructure, and (2) overlapping subproblems**
- **Bottom-up sub-chain optimization via composition from solved subproblems**
- **Top-down read-out of optimal split matrix**



Task 6 Data Access Optimizations

- **Task 6a: Describe min-max quantization of an FP64 (floating point) representation into UINT8 (integer). Why does such an encoding increasing training and/or inference performance? [8/100 points]**



- **Determine min/max range of matrix**
- **Split range into 2^8 = 256 buckets/bins**
- **Encode FP64 values in a bin via binID (lossy, but order-preserving)**

➔ **Performance Improvements**

- (1) Reduced memory bandwidth requirements
- (2) Increased instruction parallelism (e.g., AVX512: 8 FP64 → 64 UINT8 ops)
- (3) Reduced energy consumption

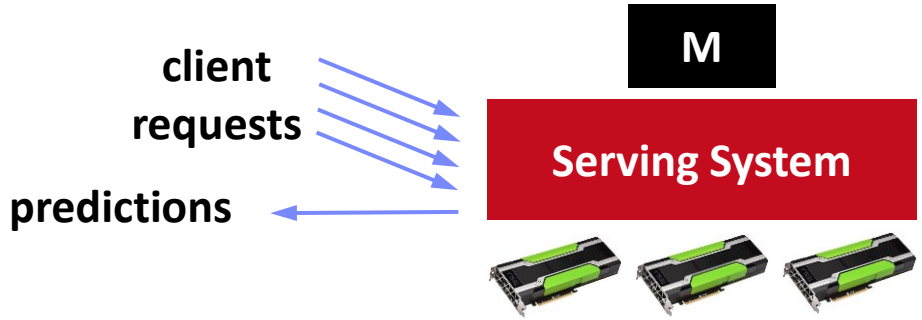
Operation	Energy
Load from **DRAM**	640 pJ
Load from **large SRAM**	50 pJ
Move 10mm across chip	32 pJ
Load from **local SRAM**	5 pJ
64-bit FMA	5 pJ
32-bit FMA	1.2 pJ
16-bit IMUL	0.26 pJ
8-bit IADD	0.01 pJ

[Jonathan Ragan-Kelly: The Future of Fast Code: Giving Hardware What It Wants, **PLDI 2024** Keynote (inspired by Bill Dally on 14nm)]

Task 7 Model Deployment

100/100

- **Task 7a: Consider a deployed model M in a cloud serving environment and assume 1000s of clients. Explain three strategies for improving model scoring throughput at the serving site. [9/100 points]**



- **Reuse of input-prediction pairs (fewer model invocations)**
- **Batching of client requests / vectorization**
  (fewer kernel launches, utilize compute better, less sync barriers)
- **Quantization of input data (data transfer to serving systems, instruction parallelism)**
- **Inference Program Compilation**
- **Specialized, Smaller Models**

THANKS
and
GOOD LUCK!